Programming for Security (5):
Yoder and Barcalow

by Sasha Romanosky and M. E. Kabay

[A note from Mich: Reader Sasha Romanosky of Morgan Stanley sent me a stimulating letter as a followup to the series on programming and security; he has very kindly allowed me to share it with readers. The following is an edited version of his original letter. In what follows, the first person refers to Mr Romanosky.]

***


The authors took the premise of OO design patterns and applied it to security. They introduced the following patterns:

* Single Access Point: preventing back doors by forcing a single entry point to code.

* Check Point: Organizing security checks and the repercussions of security violations.

* Roles: Organizing role-based security to define security privileges for different job functions.

* Session: Localizing global information about users, their privileges, resources in use and application states (e.g., locking).

* Limited View: Allowing users to see only the functions and fields that they can access.

* Full View with Errors: Showing users a full view of all functions fields (but not contents) with disabled functions and inaccessible fields clearly marked.

* Secure Access Layer: Integrating application security with low-level security such as encryption, firewalls, and authentication methods.

The paper excited me because it seemed like a great way to organize the concepts and practices that should make up a good application-security policy. One takes existing or desired practices and formulates them into security patterns. When one needs to implement a new application, host or network, one can quickly identify the security patterns from this collection of best-practice implementations and apply them to the new application design.

Collecting and formalizing known security principles in this way is of great value in developing and applying good security measures.
Typically, security measures seem to focus on network security and rarely tackle security at the application level. The authors, I believe, attempt to fill this gap. I'll note, however, that many of these patterns can (and happily do) apply to both network and applications.

Since the article, I have been working to extend their list with additional patterns. I have developed eleven so far.

If readers are interested, I would encourage you to read the original paper. If you are still interested, I would welcome your insight or the opportunity to exchange thoughts. I would like to consider this work very collaborative and should be technology and company agnostic. You may contact me at <Sasha.Romanosky@morganstanley.com>.

***

Sasha Romanosky is a senior security engineer at Morgan Stanley.

M. E. Kabay, PhD, CISSP is Associate Professor in the Department of Computer Information Systems at Norwich University in Northfield, VT. Mich can be reached by e-mail at <mkabay@compuserve.com>. He invites inquiries about his information security and operations management courses and consulting services.

Copyright © 2001 Sasha Romanosky and M. E. Kabay. All rights reserved.

Permission is hereby granted to Network World to distribute this article at will, to post it without limit on any Web site, and to republish it in any way they see fit.